Belloc, M., Velastin, S.A., Fernández, R. y Jara, M. (2018). Detection of People Boarding/Alighting a Metropolitan Train using Computer Vision. In *9th International Conference on Pattern Recognition Systems*.

DOI: https://doi.org/10.1049/cp.2018.1281

Detection of People Boarding/Alighting a Metropolitan Train using Computer Vision

M. Belloc*, S.A. Velastin+, R. Fernandez, M. Jara++**

*University Paul Sabatier (France), mathieu.belloc2@gmail.com
+Universidad Carlos III de Madrid (Spain), Cortexica Vision Systems Ltd. (UK)
and Queen Mary University of London (UK), sergio.velastin@ieee.org
**Universidad de los Andes (Chile), rfa@miuandes.cl
++Universidad de Santiago de Chile (Chile), miguel.jara.rodriguez@gmail.com

Keywords: Pedestrian Detection, HOG, Support Vector Machine, Deep Learning

Abstract

Pedestrian detection and tracking have seen a major progress in the last two decades. Nevertheless there are always application areas which either require further improvement or that have not been sufficiently explored or where production level performance (accuracy and computing efficiency) has not been demonstrated. One such area is that of pedestrian monitoring and counting in metropolitan railways platforms. In this paper we first present a new partly annotated dataset of a full-size laboratory observation of people boarding and alighting from a public transport vehicle. We then present baseline results for automatic detection of such passengers, based on computer vision, that could open the way to compute variables of interest to traffic engineers and vehicle designers such as counts and flows and how they are related to vehicle and platform layout.

1 Introduction

Most countries are trying to introduce measures to reduce the exclusive use of private means of transport by providing attractive public transport alternatives so as to reduce congestion, pollution and their associated costs. Public transport therefore needs to be seen by the general public as safe, secure and efficient. In the case of buses and railways, an aspect that limits their efficiency is the time they have to spend at stops waiting for people to get off and on. That time depends on a number of factors including cultural aspects but above all on the design of the vehicles (e.g. door width, number and height of steps, internal layout) and the stop (e.g. horizontal and vertical gaps between vehicle and platform, width, payment scheme). Especially in developing countries, the design of vehicles and stops are fixed by foreign manufacturers and building norms that are not necessarily fine-tuned to local conditions. So it becomes interesting to study the behaviour of passengers and its relation to the public transport environment. This can be done on-site, but traditionally it involves significant amount of human observation effort and of course it is not possible to vary the existing layout to find alternative designs that minimise dwell times (the time it takes a vehicle to load and unload passengers and proceed to the next stop). On the other hand, computer-based models are difficult to produce to replicate human behaviour that is largely unknown. Therefore, a suitable approach is to use full-scale vehicle/stop replicas with real people to measure dwell times (and also other variables of interest such as mean flows in/out) and their relation to layout variables. It might be argued that it is not possible to replicate operational conditions even with full-size models, because human participants might be aware that they are in under observation, but it has been shown [1] that after an initial period of getting used to an experimental set-up, participants behave in similar ways as they would in normal conditions. The laboratory known as PAMELA at University College London [2] is an advanced facility to carry out this kind of experiments where people movements are captured by video cameras for later analysis. Even then, the analysis has been conventionally done manually and requires a great deal of human time and effort. In this paper we first present a new dataset of video recordings and manual annotations (of people's head locations and sizes) to allow future researchers to investigate passenger detection, tracking, counting etc., associated with this transport engineering problem. Then, we present baseline results on the use of computer vision techniques (both involving hand-crafted features and also with deep learning training), again to allow other researchers to measure improvements over the results presented here. We focus here on the problem of detecting people in such environments, as that is the basis for then counting how many get in/out within a given period of time. Space limitations prevent us from presenting all results, something that we plan to do in a longer paper.

2 Related Work

2.1 People detection

The first step to count people getting on/off a Metro is to detect them accurately, even under crowded or semi-crowded conditions. Since the environment is mostly static we review works that match this case study. In [3] the authors use an adaptive

mixture of Gaussians to model the background. Since this kind of processing is time consuming and pedestrian detection must be real-time, authors decided to update the background model only when the number of detected persons is under a threshold. This approach relies on the hypothesis that illumination changes in an underground station are relatively slow. By using this trick, processing time is considerably reduced.

In [4] authors point out that background modelling is not suitable when the scene is completely static, for example when pedestrians are waiting to enter the metro. To tackle this problem they propose to model small motions like people turning around or moving their heads. The probability of motion occurrence is predicted from colour changes between two consecutive frames, using MID (Mosaic Image Difference) features.

Another approach to detecting heads is to use multiple cameras. By calculating the homography between the planes of each image from two different cameras and projecting the second image in the plane of the first one, a probability map can be obtained by processing the variance between the first image and the projected second image. The lowest variance area locates the head of the pedestrian. This approach is described in more detail in [5]. In our case it is not possible to use multiple cameras. However, head detection is still a good solution to tackle the occlusion problem when using background subtraction. As mentioned in [6] the foreground object extracted by background subtraction deviates from true density when the crowd is dense. To avoid this problem, they try to define regions of interest by getting gradient information that is used to approximately locate the head areas. Then, a background subtraction is applied and sub-windows are placed in the region of interest to calculate integral channel features as in [7]. The authors show that when using LUV images, heads have the highest response in the U channel.

2.2 Tracking

So as not to count the same pedestrian more than once, we need to track each new detection until it disappears from the scene. To do so, the classic Kalman filter tracking method is not suitable to our case study as it is difficult to model pedestrian behaviour with linear dynamics models. Another problem with Kalman filters is the data association step, which is difficult to achieve when appearances are similar. Most current methods use on-line learning of features for re-identifying (temporal association) pedestrians. Usually colour and shape features are used to recognize a person [8] [9] [10], as clothes' colour is the most dominant and discriminant feature in such a surveillance scene. Regarding shape features, authors usually consider a front view pedestrian to learn distinctive features, which is not our case. A novel approach consists in using DTW (Dynamic Time Warping) to use a distance measurement between two pedestrian observations for re-identification [11].

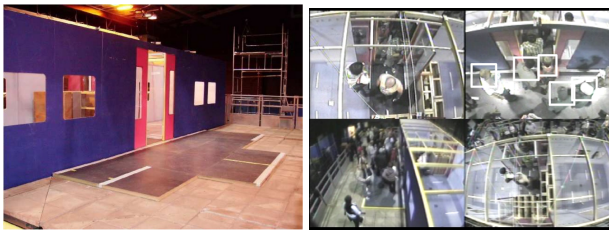
In [12] the authors propose to oversample the background to give more context information to the tracker. This goes in the opposite way to recent approaches for tracking purpose such as the ones proposed in [13] [14] [15]. The oversampling of negative samples is made possible, without degrading time performance, by using circulant matrices that allows not to increase the algorithm complexity when giving more samples to process in the Fourier domain.

Other tracking methods of the state of the art such as TLD (Tracking Learning Detection) [14] and Struck [16] are not suitable for our application because they are respectively too slow and showed poor results.

3 PAMELA-UANDES Dataset

As explained in [17] the video recordings for the dataset used in this work, were carried out in October 2008, simulating a metro train carriage constructed so as to simulate a London Underground train (Figure 1a) and the experiment used British volunteers. The variables looked at were (the video dataset contains sequences for each case):

1. Door width (800 and 1600 mm)
2. Height difference between the platform and the train (0, 150 and 300 mm)
3. Payment system (to emulate a bus), either payment to a driver or via a card



(a) (b)

Figure 1. (a) PAMELA configured as a London Underground carriage [17], (b) The different camera views (the top right also illustrates typical annotations)

The different camera views that make up the dataset are shown in Figure 1b. For the work reported here we concentrate on the view shown on the top right as this is the one more likely to be available in an operational environment. The dataset and ground truth can be downloaded from http://videodatasets.org/PAMELA-UANDES.

The working hypothesis is that head shapes are discriminating enough and once people are located they could be tracked from frame to frame to lead to measurements of counts and flows. In this paper we concentrate on the problem of head locations as its accuracy will determine how good the rest of the measurements are.

The annotated dataset consists of 8 videos or people Alighting (A_d800mm_R1..8.mpg) and 7 videos of people waiting and then Boarding (B_No_d800mm_R1..7.mpg) the full-size metro model (door width 800mm, horizontal and vertical gaps of zero). These videos last between 1 and 2 minutes and have a 352x288 resolution at 25 frames per second. This is only a subset of all the recordings made, as it takes significant effort to manually annotate each frame (it took one of the authors about 2 months to annotate this dataset), even when using the ViPER

tool [18]. To generate the ground truth data, in each frame each pedestrian head was located manually and a rectangle enclosing head and shoulder was annotated. Following findings by Dalal et al [19], the rectangles are expanded by 20% to define positive samples. Negative samples were then extracted randomly from each frame (the same negative samples are used in all the experiments) so that they do not overlap the positive samples. This resulted in a total of 43,751 positive samples and 65,625 negative training samples extracted from videos A_d800mm_R1..4.mpg and videos B_No_d800mm_R1..4.mpg, used for training. For testing, there are 41,533 positive and 62,298 negative samples from videos A_d800mm_R5..8.mpg and B_No_d800mm_R5..7.mpg

4 Computer Vision Analysis

This section describes the processes we have applied to the PAMELA-UANDES annotated dataset to obtain the baseline results reported here. The source code and results can be found in https://github.com/velastin/UAndes where interested readers can get all parameter settings we have used.

4.1 Classification

The training data is first used to train a head detector based on HOG (Histogram of Oriented Gradients) features [19] and a linear SVM (Support Vector Machine) classifier. The samples are size normalized to 56x56 (slightly larger that the maximum annotated size) using bilinear interpolation. This forms the basis to obtain a baseline with a well-established method. We refer to evaluation of *classification* as the process of measuring the performance of the output of the feature extractor-classifier combination for a given candidate image that might contain either a single head or no head (i.e. to a candidate image of the same size as used for training, in our case 56x56). This is a conventional binary classification problem. In this case, evaluating for the testing ground truth, for the HOG-SVM combination we obtain a precision (P) of 97.95% a (R) recall of 97.37% and an $F1$ of 97.66%, where precision, recall and $F1$ are as defined in eqs. 1 and 2.

$$P = TP/(TP + FP), R = TP/(TP + FN) \quad (1)$$

$$F1 = 2\frac{PR}{P + R} \quad (2)$$

where TP is the number of true positives (correctly detected heads), FP is the number of false positives (non heads detected as heads) and FN is the number of false negatives (heads incorrectly classified as non-heads). We also wanted to test the behaviour of newer classification methods that use deep learning and so we tried a fine-tuned version of Inception V3 [20] as a classifier. This deep neural network was trained in two phases: first we froze all the layers weights and trained only the 3 classification layers (global spatial average pooling layer, fully connected layer and logistic layer). Then we decided to unfreeze the layers corresponding to the top 2 blocks of the Inception V3 architecture to fine-tune the weights of these layers and make the network fit the dataset better. Achieving this training part we obtained 99.31% accuracy (eq. 3) on a set of 10000 samples from the test ground truth.

$$A = (TP + TN)/(TP + FN + TN + FP) \quad (3)$$

4.2 Detection

The really interesting problem to solve is that of localising pedestrians in a previously unseen image. We refer to this as the *detection* problem. The conventional approach is to sweep a sliding window (of the same size as used for training, so 56x56 in our case) through every possible position in the image. In an extreme case, the sliding window is moved a pixel at a time (horizontally and vertically). If the objects to detect are expected to have a range of sizes, it is necessary to sweep the sliding window at different scales (in our case the range is small enough to avoid this). Clearly, a detection process using "full image" sliding window is bound to be time consuming. An alernative is to identify areas in the images which are more likely to contain pedestrians, and that therefore we will call here *Candidate Regions* (CRs). Then, the sliding window is only applied to the CRs and not the whole image, saving computation time and even reaching real-time. We consider two ways to do this.

First, we use a background subtraction based on MOG (Mixture of Gaussians) as described in [21], [22]. This background model allows us to extract moving objects but we cannot rely only on this method to detect pedestrians directly, as it only generates binary images with foreground pixels. Moreover, background subtraction has difficulty in segmenting foreground objects when pedestrians stop moving because pixel weights associated to those foreground areas will decrease until falling under the "foreground-background" threshold and then become background [4]. Another drawback of background subtraction is the difficulty to identify moving foreground objects when pedestrians wear clothes of similar colour as the background. Because we are not aiming to segment the foreground objects perfectly we extract larger rectangular Candidate Regions (CRs) than the detected foreground areas, see Figure 2, so as not to reduce the chance of missing pedestrians. Clearly, a single CR corresponding to the full image corresponds to sweeping the full image.

Secondly, we implemented a naive frame difference method that computes "foreground" (motion) pixels from thresholded (>30) frame-frame absolute differences.

For each of the detected CRs a 56x56 sliding window with a step of 8 pixels on the x and y axes is applied. Each extracted descriptor (HOG for SVM and the 56x56 raw image for Inception) is passed to the corresponding classifier. This generates a number of candidate heads. As is inherent with a sliding window, it is possible that multiple "hits" (candidate heads) are generated for a single pedestrian. To reduce extra hits, we apply a Non Maxima Suppression (NMS) process on the basis of the confidence in each detection to retain only detections that have the highest chance to be a pedestrian head see Figure 3.

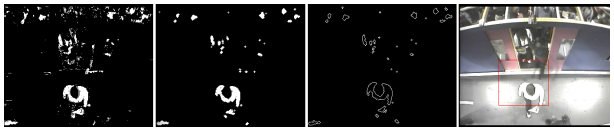


Figure 2. Candidate Regions selection process. From left to right: Pixels set as foreground (white) or background (black), Apply morphological opening with a 5x5 cross kernel to reduce noise and to minimise shadows linking people (especially when crowd is denser), Extract contours, Inscribe contours with rectangular CRs (ignoring those below a given size).

Figure 3. Pedestrian detection (for simplicity only one head is shown). Left: detected head found in the CR shown in Figure 2, Right: highest confidence detection retained after NMS

4.3 Detection Evaluation

Evaluating detection is more complicated than evaluating just classification (sec. 4.1), because we need to deal with possible shifts and scale changes with respect to the ground truth in the detected objects and also multiple detections for any given ground truth. We have chosen the commonly used Jaccard coefficient [23]. Given a rectangle of pixels R_g corresponding to a head in the ground truth and another rectangle of pixels R_d resulting from the detection process, the degree of similarity J (effectively a measure of overlap) is given by eq. 4 (union and intersection are simply defined in terms of areas, in pixels). Then a detected object whose $J{>}\tau$ is for a ground truth object is considered a true positive. A detected object for which $J{<}\tau$ for all ground truths in that frame is considered a false positive and a ground truth that has no detected object satisfying $J{>}\tau$ is a false negative (missed detection).

$$J = \frac{R_g \cap R_d}{R_g \cup R_d} \tag{4}$$

Table 1 shows detection results with various methods. "Normal" training refers to conventional one-pass SVM training with positive and negative samples while "hard" training [19] is one where a second training is carried out with additional negative samples that have been identified to generate false positives with the first training. Mean and variances are calculated from the results for each test video and as their nature is fairly similar no large variations have been observed. The Jaccard threshold τ is set to 10%, given that the average size of a head is around 20x20 pixels so this corresponds to an effective overlap of a bit less than 50%. The NMS search area is set to 10% of the detection window. Detected ratio is defined as the number of true positives divided by the number of ground truth positives. Note that this is not the same as recall because of the way that true positives have been calculated by the evaluation framework we have used, where multiple detections of the same ground truth positive are counted as multiple true positives. This can explain the decrease in recall after NMS, because NMS reduces the number of multiple true positives. This effectively means that it is not too useful to compare different columns but to see these results as giving an indication of the relative differences between the different methods (within a column). In that context and focusing on $F1$ (the combination of precision and recall), it is clear that the methods based on the deep learning inception architecture, outperform the conventional HOG-SVM combination. Finally, we also note that the mean speed ups (compared to using full frames) are 6.7 and 8.1 for MOG and interframe difference respectively, so although full frame $F1$ figures are slightly better, the significant saving in computing times could be attractive.

4.4 Detection by tracking

As explained earlier, one of the end requirements is to measure flows (e.g. the number of people crossing through the door over a certain period of time) and that would require tracking/re-identification of individuals from frame to frame. Furthermore, exploiting temporal consistency could improve detection itself (e.g. by removing unlikely detections). In this section we explore two alternative trackers: KCF (Kernelized Correlation Filter) [12] and a simpler colour histogram "tracker" (which is twice as fast as the KCF tracker). In the latter case, for each detection that does not share more than 10% of common area with any already tracked region, a new tracker is instantiated with an initial reference colour histogram extracted from the 56x56 detection window. Then the pedestrian detection process is repeated for the next frame. If one of the new detections generates a Jaccard coefficient greater than a threshold with an already tracked region we compute the histogram distance using Hellinger method. If the distance is less than 1.5*mean distance of all previously calculated distances for this track, the tracker is updated with this new detection and the track sequence is iteratively saved like this. On the other hand, if none of the new detections satisfies the such conditions we compute the distance travelled in the last 2 tracked regions. Assuming that the speed is constant between 2 frames and that direction remains the same, we estimate an approximative location of the pedestrian in the current frame. To re-detect it, we apply the sliding window in a 96x96 region centred on the estimated point. This process allows us to recover from small target loss that may happen when the person detector miss-classifies a pedestrian or when the CR selection method fails.

To detect the termination of a track (the target is out of the image bounds or is lost) we count the number of consecutive frames for which a tracker could not be updated by either measurement or re-detection. If this number exceeds 5 consecutive frames the track is removed and the track sequence is saved, provided it was longer than 20 consecutive frames.

Concerning the KCF tracker we rely on its prediction when

Table 1. Detection results

Method	Mean/var	Precision	Recall	F1	detected ratio	precision after NMS	recall after NMS	F1 after NMS	detected ratio
HOG+SVM:									
MOG	mean	66.25%	61.22%	63.43%	31.14%	81.14%	18.80%	30.38%	18.91%
(normal training)	var	0.20%	0.28%	0.10%	0.16%	0.29%	0.09%	0.14%	0.09%
frame diff	mean	67.26%	59.32%	62.19%	28.89%	83.35%	18.31%	29.40%	18.39%
(normal training)	var	0.12%	1.73%	0.49%	0.86%	0.05%	0.54%	0.97%	0.54%
full frame	mean	49.48%	**97.29%**	64.96%	89.03%	77.38%	48.34%	59.45%	48.47%
(normal training)	var	1.31%	0.01%	1.00%	0.16%	0.60%	0.15%	0.24%	0.16%
frame diff	mean	58.94%	71.30%	63.99%	32.57%	80.41%	16.71%	27.09%	16.81%
(hard training)	var	0.15%	1.28%	0.17%	0.77%	0.09%	0.53%	1.02%	0.53%
Inception v3:									
MOG	mean	80.76%	90.05%	85.09%	45.68%	84.78%	36.24%	50.62%	36.43%
	var	0.13%	0.06%	0.04%	0.25%	0.07%	0.18%	0.15%	0.18%
frame diff	mean	**82.52%**	77.06%	79.55%	27.50%	**84.96%**	23.22%	35.72%	23.30%
	var	0.10%	0.91%	0.41%	0.68%	0.05%	0.77%	1.18%	0.77%
full frame	mean	75.85%	98.74%	**85.76%**	**89.15%**	71.93%	**70.62%**	**70.65%**	**70.78%**
	var	0.11%	0.00%	0.05%	0.06%	1.24%	0.19%	0.17%	0.19%

Table 2. Detection by tracking

Method	Mean/Var	precision after tracking (colour)	recall after tracking (colour)	F1 after tracking (colour)	detected ratio	precision after tracking (KCF)	recall after tracking (KCF)	F1 after tracking (KCF)	detected ratio
MOG	mean	49.48%	97.29%	64.96%	89.03%	77.38%	**48.34%**	59.45%	**48.47%**
(normal training)	var	1.31%	0.01%	1.00%	0.16%	0.60%	0.15%	0.24%	0.16%
frame diff	mean	67.11%	63.23%	65.11%	25.99%	83.18%	10.09%	17.99%	10.44%
(normal training)	var	60.65%	61.42%	61.03%	26.12%	78.30%	9.73%	17.31%	9.77%
full frame	mean	76.63%	62.66%	**68.94%**	16.49%	82.51%	12.65%	21.94%	12.68%
(normal training)	var	81.93%	72.07%	76.68%	21.87%	89.46%	15.70%	26.71%	16.02%
frame diff	mean	80.76%	90.05%	85.09%	45.68%	84.78%	36.24%	50.62%	36.43%
(hard training)	var	0.13%	0.06%	0.04%	0.25%	0.07%	0.18%	0.15%	0.18%
Inception v3:									
MOG	mean	**92.85%**	31.03%	46.41%	28.86%	**90.47%**	47.09%	**61.83%**	44.24%
	var	0.07%	0.11%	0.14%	0.07%	0.05%	0.14%	0.09%	0.12%
frame diff	mean	89.62%	23.10%	36.44%	20.76%	89.49%	35.60%	50.70%	33.08%
	var	0.24%	0.39%	0.66%	0.40%	0.08%	0.38%	0.40%	0.48%
full frame	mean	82.86%	**61.31%**	70.37%	**55.42%**				
	var	0.68%	0.44%	0.47%	0.50%				

we find a Jaccard coefficient greater than 0.5 with the new detections otherwise we re-initialize it with the re-detected object using the same method as for colour histogram.

Table 2 shows how detection results are updated by tracking (for technical reasons it was not possible to produce full frame inception results). These results need to be compared with the "after NMS" columns in table 1. Focusing on $F1$ (as it is easier to use a combined metric) we see that the colour histogram tracker tends to improve the results obtained for HOG-SVM. For inception, that tracker does not make a significant difference. For KCF with HOG-SVM detection results are poorer than for the simpler colour histogram and do not show overall improvement to detection on its own. Interestingly, the converse is true with inception-based detection and that combination resulting in the best after NMS results overall (an $F1$ of 61.83%).

5 Conclusion

In this work we have presented a new public realistic and challenging dataset for pedestrian detection. We have also provided baseline results for future researchers to improve upon. Using both well-known and newer approaches, we have shown that although it is possible to detect and track most of the people present in the dataset, there is still much to do before such algorithms could be robustly used in a final application. We are currently refining the performance evaluation methodology to allow better comparisons of results and also considering other deep-learning based methods to assess their performance on this type of environment. Future work is expected to consider improvements in tracking and exploiting motion information to refine detection. Although it is difficult to capture and annotate new data (e.g. for another railway operator, in shopping malls), it would be useful to check if what is proposed here is sufficiently generic to be used elsewhere. We are also studying more closely how to explicitly deal with occlusion and exploring the use of pose estimation (e.g. using RGB-D sensors) to improve detection and hence tracking and counting.

Acknowledgements

The authors gratefully acknowledge the Chilean National Science and Technology Council (Conicyt) for its funding under grants CONICYT-Fondecyt Regular nos. 1140209 ("OBSERVE") , 1120219, and 1080381 . S.A. Velastin is grateful to funding received from the Universidad Carlos III de Madrid, the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 600371, el Ministerio de Economía y Competitividad (COFUND2013-51509) and Banco Santander. Finally, we are grateful to NVIDIA for its donation as part of its academic GPU Grant Program.

References

[1] R. Fernández, P. Zegers, G. Weber, and N. Tyler, "Influence of platform height, door width, and fare collection on bus dwell time: laboratory evidence for santiago de chile," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2143, pp. 59–66, 2010.

[2] UCL Transport Institute, "PAMELA." http://www.ucl.ac.uk/transport-institute/Research-snapshots/PAMELA, accessed 2017-02-06.

[3] X. Hu, H. Zheng, W. Wang, and X. Li, "A novel approach for crowd video monitoring of subway platforms," *Optik-International Journal for Light and Electron Optics*, vol. 124, no. 22, pp. 5301–5306, 2013.

[4] M. Li, Z. Zhang, K. Huang, and T. Tan, "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1–4, IEEE, 2008.

[5] R. Eshel and Y. Moses, "Homography based multiple camera detection and tracking of people in a dense crowd," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[6] V. B. Subburaman, A. Descamps, and C. Carincotte, "Counting people in the crowd using a generic head detector," in *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pp. 470–475, IEEE, 2012.

[7] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," 2009.

[8] D.-N. T. Cong, L. Khoudour, C. Achard, and L. Douadi, "People detection and re-identification in complex environments," *IEICE TRANSACTIONS on Information and Systems*, vol. 93, no. 7, pp. 1761–1772, 2010.

[9] C. Coniglio, C. Meurie, O. Lézoray, and M. Berbineau, "A graph based people silhouette segmentation using combined probabilities extracted from appearance, shape template prior, and color distributions," in *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 299–310, Springer, 2015.

[10] C.-H. Kuo and R. Nevatia, "How does person identity recognition help multi-person tracking?," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1217–1224, IEEE, 2011.

[11] D. Simonnet, M. Lewandowski, S. Velastin, J. Orwell, and E. Turkbeyler, "Re-identification of pedestrians in crowds using dynamic time warping," in *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pp. 423–432, Springer, 2012.

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[13] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *European Conference on Computer Vision*, pp. 864–877, Springer, 2012.

[14] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[15] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[16] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr, "Struck: Structured output tracking with kernels," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2096–2109, 2016.

[17] R. Fernandez, "Experimental study of bus boarding and alighting times," in *European Transport Conference, Glasgow*, pp. 10–12, 2011.

[18] D. Doermann and D. Mihalcik, "Tools and techniques for video performance evaluation," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 4, pp. 167–170, IEEE, 2000.

[19] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

[21] Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern recognition letters*, vol. 27, no. 7, pp. 773–780, 2006.

[22] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, pp. 28–31, IEEE, 2004.

[23] M. Levandowsky and D. Winter, "Distance between sets," *Nature*, vol. 234, no. 5323, p. 34, 1971.